

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR UNITED STATES PATENT

FOR

**SYSTEM, DEVICE, AND METHOD FOR MANAGING  
SERVICE LEVEL AGREEMENTS  
IN AN OPTICAL COMMUNICATION SYSTEM**

Inventors:

**Indermohan S. Monga**  
217 Massachusetts Avenue  
Lexington, MA 02420

**Bruce A. Schofield**  
15 Farwell Road  
Tyngsboro, MA 01879

**Rajesh Tyagi**  
21 Hibiscus Lane  
Ayer, MA 01432

Attorney Docket No.: 2204/A98

Client Reference No.: 13072BAUS08U

Attorneys:

**BROMBERG & SUNSTEIN LLP**  
125 Summer Street  
Boston, MA 02110  
(617) 443-9292

0930375 081501  
"5/2008"

**SYSTEM, DEVICE, AND METHOD FOR MANAGING  
SERVICE LEVEL AGREEMENTS  
IN AN OPTICAL COMMUNICATION NETWORK**

5

**PRIORITY**

The present application claims priority from the following commonly-  
owned United States patent application, which is hereby incorporated herein  
10 by reference in its entirety:

United States Provisional Patent Application No. 60/225,335 entitled  
**OPTICAL SERVICE AGENT**, which was filed on August 15, 2000 in the  
names of Bruce A. Schofield, William R. Hawe, Paul D. Callahan, Indermohan  
Monga, Stephen Suryaputra, and Andre N. Fredette.

15

**CROSS-REFERENCE TO RELATED APPLICATION(S)**

The present application may be related to the following commonly-  
20 owned United States patent applications, which are hereby incorporated  
herein by reference in their entireties:

United States Patent Application No. XX/XXX,XXX entitled **OPTICAL  
SWITCH ROUTER**, which was filed on even date herewith in the names of  
Bruce A. Schofield, James V. Luciani, and Michael J. Craren;

25

United States Patent Application No. XX/XXX,XXX entitled **SYSTEM  
AND DEVICE FOR PROVIDING COMMUNICATION SERVICES IN AN  
OPTICAL COMMUNICATION SYSTEM**, which was filed on even date  
herewith in the names of Bruce A. Schofield, Indermohan S. Monga, and  
Stephen Suryaputra; and

30

United States Patent Application No. XX/XXX,XXX entitled **SYSTEM,  
DEVICE, AND METHOD FOR MANAGING COMMUNICATION  
SERVICES IN AN OPTICAL COMMUNICATION SYSTEM**, filed on even  
date herewith in the names of Bruce A. Schofield, William R. Hawe, Paul D.  
Callahan, Indermohan S. Monga, Stephen Suryaputra, and Andre N. Fredette;

35

and

United States Patent Application No. XX/XXX,XXX entitled **SYSTEM,  
DEVICE, AND METHOD FOR MANAGING CONNECTION  
ESTABLISHMENT AND RELATED SERVICES IN AN OPTICAL  
COMMUNICATION SYSTEM**, filed on even data herewith in the names of  
5 Indermohan S. Monga, Stephen Suryaputra, Vikram Ramachandran, and  
Bruce A. Schofield.

### **FIELD OF THE INVENTION**

10 The present invention relates generally to optical networking, and  
more particularly to managing communication services in an optical  
communication system.

### **BACKGROUND OF THE INVENTION**

15 In today's information age, communication networks are increasingly  
used for transferring information among a multitude of communication  
20 devices. As demand for communication services continues to grow, the  
demand on these communication networks for carrying increasing amounts  
of information at increasing speeds continues to grow. Therefore,  
communication networks are evolving to handle these increased demands.

25 One way in which communication networks are evolving is through  
the use of optical communication facilities. Optical communication facilities  
transport information over optical communication media (such as optical  
fibers). Such optical communication networks provide vast amounts of  
bandwidth over short and long distances.

30 The optical communication network is capable of providing various  
communication services to its users. Traditionally, such communication  
services have been very expensive, and need to be planned and scheduled  
well in advance. This is due in part because changes to the optical

communication network, such as provisioning and switching optical communication paths, required substantial human intervention.

## SUMMARY OF THE INVENTION

In accordance with one aspect of the present invention, an optical service agent manages service level agreements for a user. The optical service agent can perform both real-time and off-line analysis for the user, and can interact with various network elements (including the core optical communication network) to handle billing, penalty, and other issues associated with a SLA breach. Among other things, the optical service agent may monitor and analyze a connection in real-time for determining SLA compliance, gather and maintain statistical information relating to a connection, analyze the statistical information off-line for determining SLA compliance, patterns, and trends, interact with a service provider to enforce penalty provisions in the SLA, interact with a service provider to negotiate a credit for services not provided by the service provider in accordance with the SLA, interact with a service provider to negotiate "replacement" services for a breach of the SLA, interact with various network elements to rectify a breach of the SLA, interact with the service provider to dynamically modify the SLA based upon changing user requirements, and interface with a billing/accounting system to provide SLA-related information. The optical service agent can also interact with various management components in the optical communication network.

## BRIEF DESCRIPTION OF THE DRAWINGS

In the accompanying drawings:

FIG. 1 is a network diagram showing a representation of a communication system in which users communicate over an automatically

switched optical network (ASON) in accordance with an embodiment of the present invention;

FIG. 2 is a network diagram showing an exemplary ASON in accordance with an embodiment of the present invention;

5 FIG. 3 is a block diagram showing relevant components of an exemplary ASON device in accordance with an embodiment of the present invention;

10 FIG. 4 is a block diagram showing an ASON-enabled user including an ASON-enabled user application in accordance with an embodiment of the present invention;

FIG. 5 is a network diagram showing an exemplary communication system in which ASON-enabled users communicate over an ASON in accordance with an embodiment of the present invention;

15 FIG. 6 is a block diagram showing an OSA-enabled user including an embedded OSA in accordance with an embodiment of the present invention;

FIG. 7 is a diagram showing the relationship between a user application and the OSA in accordance with an embodiment of the present invention;

20 FIG. 8 is a diagram showing the relationship between the OSA-A and the OSA-N in accordance with an embodiment of the present invention;

FIG. 9 is a diagram showing an exemplary system in which the OSA-N resides in the OSA-enabled user in the edge system in accordance with an embodiment of the present invention;

25 FIG. 10 is a diagram showing an exemplary system in which the OSA-N resides in the ASON user at the edge of the ASON in accordance with an embodiment of the present invention;

FIG. 11 is a diagram showing an exemplary system in which the OSA-N resides outside of the OSA-enabled user and the ASON device in a proxy arrangement in accordance with an embodiment of the present invention;

30 FIG. 12 is a diagram depicting how OSA-N functionality is handled by the OSS in accordance with a client-server embodiment of the present invention;

FIG. 13 is a message flow diagram depicting various exchanges between the OSA-N and the OSS in accordance with a client-server embodiment of the present invention;

5 FIG. 14 is a diagram depicting how authentication is handled by the OSA in accordance with a distributed flooding embodiment of the present invention;

FIG. 15 is a message flow diagram depicting various exchanges between the OSA-N and the OSS in accordance with a distributed flooding embodiment of the present invention;

10 FIG. 16 is a diagram depicting how authentication and flooding are handled by the OSS in accordance with a hybrid/proxy embodiment of the present invention;

15 FIG. 17 is a message flow diagram depicting various exchanges between the OSA-N and the OSS in accordance with a hybrid/proxy embodiment of the present invention;

FIG. 18 is a network diagram showing an exemplary communication system in which an OSA-enabled user communicates with an ASON-enabled user over the ASON in accordance with an embodiment of the present invention;

20 FIG. 19 is a network diagram showing an exemplary communication system in which two OSA-enabled users communicate over the ASON in accordance with an embodiment of the present invention;

FIG. 20 is a block diagram showing relevant components of the OSA in accordance with an embodiment of the present invention;

25 FIG. 21 is a logic flow diagram showing exemplary OSA logic for managing communication services in accordance with an embodiment of the present invention;

30 FIG. 22 is a message flow diagram demonstrating an authenticated auto-discovery process in accordance with an embodiment of the present invention;

FIG. 23 is a message flow diagram demonstrating the process of removing an OSA-enabled user from a peer group in accordance with an embodiment of the present invention;

FIG. 24 is a logic flow diagram showing exemplary OSA logic for managing a service level agreement on behalf of a network user in accordance with an embodiment of the present invention;

FIG. 25 is a logic flow diagram showing exemplary OSA logic for verifying service level agreement and identifying breaches of the service level agreement in accordance with an embodiment of the present invention; and

FIG. 26 is a logic flow diagram showing exemplary OSA logic for managing the service level agreement upon detecting a breach of the service level agreement in accordance with an embodiment of the present invention.

#### **DETAILED DESCRIPTION OF A PREFERRED EMBODIMENT**

A SLA is essentially an agreement by a service provider to provide a specific level of service to the network user. The SLA is typically negotiated between the network user and the service provider prior to, or at the time of, establishing a connection. The SLA may provide for such things as connection bandwidth, connection quality (e.g., error rate, dropped packet rate, latency), connection duration, protection class, and protection/failover time, to name but a few. The SLA typically provides for various penalties in case the service provider is unable to provide the level of service agreed to in the SLA.

In typical "buyer beware" fashion, the network user is generally responsible for identifying any breaches of the SLA and invoking the penalty provisions of the SLA. Some breaches of the SLA are obvious, such as a complete failure of a connection. However, other breaches of the SLA are less obvious. Thus, the network user must typically police the SLA in order to identify breaches of the SLA. This is typically done manually, for example, by

monitoring the connection or analyzing statistical information maintained by the network user.

Therefore, in an embodiment of the present invention, an intelligent agent operating with the domain of the network user manages service level agreements (SLAs) on behalf of the network user. For example, the SLA agent may monitor the connection in real-time in order to verify SLA, gather and maintain statistical information for off-line analysis (e.g., rejected calls, missing packets), interface with a billing/accounting system to provide SLA-related information, interact with the service provider to enforce penalty provisions in the SLA, interact with the service provider to rectify a breach of the SLA, and interact with the service provider to dynamically modify the SLA based upon changing requirements of the user, to name but a few. The SLA agent may interact with various management components in the optical communication network.

In a typical embodiment of the present invention, the SLA agent is an optical service agent (OSA) that operates within the domain of the network user. The OSA manages communication connections on behalf of the network user. The OSA interacts with the optical communication network to obtain various communication services and manages those communication services for the network user based upon predetermined parameters defined by the network user. In essence, then, the optical communication network provides a "core" set of communication services that can be accessed by the OSA, and the OSA provides advanced communication services for the network user using the "core" set of communication services provided by the optical communication network.

In an agile transport network, various optical communication services that heretofore have been performed manually are performed automatically by "intelligent" optical communication devices within the network. In particular, the optical internet infrastructure is evolving from a statically-



provisioned SONET ring-based transport network to a more dynamically-provisioned mesh network.

One example of an agile transport network is an automatically  
switched optical/transport network (ASON or ASTN), which is described in  
the related application entitled **SYSTEM AND DEVICE FOR PROVIDING  
COMMUNICATION SERVICES IN AN OPTICAL COMMUNICATION  
SYSTEM** incorporated by reference above. The ASON typically includes  
optical cross-connect switches (OXC) that are used to form the mesh network  
and optical channel controllers (OCCs) that dynamically create, delete, and  
modify optical communication paths by controlling the OXC through a  
connection control interface (CCI). For convenience, the OXC and OCC are  
referred to hereinafter collectively as ASON devices. The ASON devices, and  
in particular the OCCs, are typically network routers that use a link-state  
routing protocol (e.g., OSPF) to distribute link attributes (such as optical  
channel availability) and a signaling protocol (e.g., MPLS or GMPLS) to  
manage optical communication paths end-to-end. These protocols enable the  
ASON to automatically and quickly create, delete, and modify optical  
communication paths. The ASON typically also includes an optical service  
server (OSS) for coordinating the various communication services provided  
by the ASON. Thus, the ASON provides more agility by moving from  
manual provisioning to automatic provisioning via switching and signaling.

In essence, then, the ASON is an optical/transport network that has  
dynamic connection capabilities. Among other things, the ASON provides for  
discovery of physical topology for optical elements, efficient use of available  
bandwidth by dynamic creation of optical communication paths, and end-to-  
end connection protection and restoration.

FIG. 1 shows a representation of a communication system 100 in which  
two users 110, 130 communicate over an ASON 120. The users 110, 130 may  
be optical switch routers, as described in the related application entitled

**OPTICAL SWITCH ROUTER** incorporated by reference above, that are positioned as edge nodes of their respective user networks for interfacing with the ASON 120. The ASON 120 provides various communication services to the users 110, 130 via the automatic servicing logic. Various ASON communication services are discussed in more detail below.

Each ASON device includes an ASON controller for supporting automated communication services within the ASON 120. Among other things, the ASON controller enables each ASON device to automatically provision, switch, and signal optical communication paths within the ASON 120. The ASON controller enables carriers and service providers to offer many value-added services to their customers.

In order for the users 110, 130 to control and monitor communication services from the ASON 120, the ASON controller provides a user-to-network interface (UNI) through which the users 110, 130 interact with the ASON controller for controlling and monitoring communication services within the ASON 120. Through the ASON UNI, the users 110, 130 can access various controllable features of the ASON 120. For example, using the ASON UNI, the users 110, 130 can request an optical communication path having certain attributes, renegotiate the attributes of the optical communication path, control switching of the optical communication path, terminate the optical communication path, and monitor operation of the ASON 120, to name but a few.

Within the ASON 120, each ASON device is able to perform various functions in support of the automated communication services. Some of these functions can be carried out by an individual ASON device, while others of these functions require coordination among multiple ASON devices. Therefore, the ASON controller provides a network-to-network interface (NNI) that enables communication between ASON devices for coordinating various communication functions. Through the ASON NNI, the various

ASON devices can exchange ASON routing information and coordinate such things as establishment and termination of optical communication paths, switching of optical communication paths, and protection and restoration of optical communication paths, to name but a few.

5

FIG. 2 shows an exemplary embodiment of the ASON 120 having four ASON devices 210, 220, 230, 240 and OSS 250. For convenience, the solid lines shown between the ASON devices 210-240 represent optical communication paths, which may be all-optical paths (lightpaths) or partially optical paths (switch-paths), and the dashed lines shown between the ASON devices 210-240 show the type of interface (UNI or NNI). The users 110, 130 interface with the ASON 120 through ASON device 210 and ASON device 240, respectively. The ASON devices 210, 240 provide a UNI to the users 110, 130, respectively, through which the users 110, 130 can control and monitor communication services provided by the ASON 120, and more specifically by the ASON controller within the ASON devices 210-240. The ASON devices 210-240 are interconnected via the NNI, and use the NNI to interoperate for coordinating various communication functions. It should be noted that the ASON NNI may utilize separate communication paths from the optical communication paths. The OSS 250 coordinates the various communication services provided by the ASON devices 210-240.

At the heart of the various automated communication services is automatic switching of optical communication paths. In order to support automatic switching of optical communication paths within the ASON 120, the ASON devices 210-240 typically include some form of optical switching logic, such as an optical/photonic switching fabric, for performing optical/photonic switching of optical communication paths. The optical switching fabric may be based upon any of a variety of optical/photonic switching technologies, including, but not limited to, Micro Electro Mechanical System (MEMS) technology, Micro Opto Electro Mechanical System (MOEMS) technology, lithium niobate technology, liquid crystal

technology, or other optical/photonic switching technology. The optical switching logic can be dynamically configured under control of the ASON controller for such things as passing optical data streams from a number of incoming optical fibers to a number of outgoing optical fibers (i.e., switching),  
5 adding an optical data stream to an outgoing optical fiber, and dropping an optical data stream from an incoming optical fiber for local processing by the ASON device, to name but a few.

FIG. 3 shows the relevant components of an exemplary ASON device  
10 300. Among other things, the ASON device 300 includes a number of incoming optical interfaces 310, a number of outgoing optical interfaces 330, optical switching logic 320, ASON controller 340, ASON UNI 350, ASON NNI 360, ASON Network Management Interface (NMI) 370, and network management/optimization elements 380. The incoming optical interface(s)  
15 310 are couplable to a number of incoming optical fibers for receiving optical data streams at various wavelengths. The outgoing optical interface(s) 330 are couplable to a number of outgoing optical fibers for outputting optical data streams at various wavelengths. The optical switching logic 320 is interposed between the incoming optical interface(s) 310 and the outgoing optical  
20 interface(s) 330 for switching optical data streams, which may include such things as passing certain optical data streams received over the incoming optical interface(s) 310 through to the outgoing optical interface(s) 330, dropping one or more optical data streams received over the incoming optical interface(s) 310 for local processing by the ASON device 300, and adding one  
25 or more optical data streams to the outgoing optical interface(s) 330, to name but a few. The ASON controller 340 automates certain communication services by controlling, among other things, the incoming optical interface(s) 310, the optical switching logic 320, and the outgoing optical interface(s) 330. The ASON UNI 350 enables a user to control and monitor communication  
30 services provided by the ASON controller 340. The ASON NNI 360 enables the ASON controller 340 within the ASON device 300 to interact with the ASON controller in other ASON devices for coordinating communication

services within the ASON 120. The ASON UNI 350 and the ASON NNI 360 are typically integral to the ASON controller 340, but are shown here separately for convenience. The ASON NMI 370 is a network management interface between the ASON controller 340 and various network management/optimization elements 380. Among other things, the ASON controller 340 provides network status information to the network management/optimization elements 380 and receives network updates from the network management/optimization elements 380 via the ASON NMI 370.

Without limitation, the ASON 120 can provide practically any communication service that has traditionally been performed manually. As described in the related application entitled **SYSTEM AND DEVICE FOR PROVIDING COMMUNICATION SERVICES IN AN OPTICAL COMMUNICATION SYSTEM** incorporated by reference above, some exemplary communication services that can be provided by the ASON 120 include "plug and play" for optical elements, modeling of optical communication paths, automatic provisioning of optical communication paths, user-requested switching of optical communication paths, automatic switching based upon service level agreement (SLA), automatic switching for protection and restoration, bandwidth management, and reporting of statistical and other information, to name but a few. It should be noted that the present invention is in no way limited to any particular communication services provided by the ASON 120.

As discussed above, the ASON UNI enables the users to control and monitor communication services provided by the ASON 120. The ASON UNI provides a standardized interface to the ASON 120, and more particularly to the ASON controller 340 in the ASON device. Conceptually, the ASON UNI provides a set of "control knobs" through which the users can control and monitor ASON communication services.

One way for a user to access ASON communication services is by implementing some or all of the ASON UNI functionality. For example, ASON UNI functionality can be integrated with a user application so that the user application can access ASON communication services. For convenience, such a user is referred to hereinafter as an ASON-enabled user, and such a user application is referred to hereinafter as an ASON-enabled user application.

FIG. 4 shows an ASON-enabled user including an ASON-enabled user application 410. The ASON-enabled user application 410 implements some or all of the ASON UNI functionality. The ASON-enabled user application 410 communicates with the ASON controller 340 in the ASON device over a UNI signaling channel 420, using the ASON UNI in order to obtain communication services from the ASON 120. The UNI signaling channel 420 may be, for example, an in-band signal carried over a SONET Data Communication Channel (DCC).

FIG. 5 shows an exemplary communication system 500 in which two ASON-enabled users 510, 520 communicate over the ASON 120. Each of the ASON-enabled users 510, 520 implement some or all of the ASON UNI, and therefore are able to monitor and control certain communication services provided by the ASON 120. For example, using the ASON UNI, the ASON-enabled users 510, 520 can establish an end-to-end optical communication path for communicating over the ASON 120. The optical communication path has certain attributes that are negotiated between the two ASON-enabled users 510, 520 and also between the ASON-enabled users 510, 520 and the ASON 120.

Unfortunately, it is not always practical to implement the ASON UNI in each user application that requires ASON communication services. Therefore, another way for the user to access ASON communication services is by employing an optical service agent (OSA) to manage communication

services for the user. As described in the related application entitled  
**SYSTEM, DEVICE, AND METHOD FOR MANAGING  
COMMUNICATION SERVICES IN AN OPTICAL COMMUNICATION**

**SYSTEM** incorporated by reference above, the OSA is an intelligent,  
5 embedded signaling agent that operates within the user at the edge of the  
ASON 120. The OSA implements application-specific services and  
intelligence as well as the ASON UNI and other mechanisms for  
communicating with the ASON controller 340 via the ASON UNI. The OSA  
can be viewed as a subsystem that understands user requirements and  
10 manages various communication services on behalf of the user to meet the  
user requirements. Specifically, the OSA interacts with the ASON 120 via the  
ASON UNI to obtain various communication services and manages that  
communication services for the user based upon predetermined parameters  
defined by the user. Architecturally, the OSA is layered above the ASON  
15 UNI, and uses the ASON UNI to manage and control ASON communication  
services provided through the ASON controller 340. In essence, then, the  
OSA provides advanced communication services for the user using the  
“control knobs” provided by the ASON UNI. For convenience, a user that  
employs an OSA for managing communication services is referred to  
20 hereinafter as an OSA-enabled user.

FIG. 6 shows an OSA-enabled user including an embedded OSA 610.  
The OSA 610 implements application-specific services and intelligence as well  
as the ASON UNI and other mechanisms for communicating with the ASON  
25 controller 340 via the ASON UNI. The OSA 610 communicates with network  
management/optimization elements 380 in the ASON 120 via a network  
management interface 620, and communicates with the ASON controller 340  
in the ASON device via the ASON UNI 630 in order to obtain communication  
services from the ASON 120, specifically by sending service requests to the  
30 ASON controller 340 and receiving service responses from the ASON  
controller 340. The OSA 610 manages the communication services for the user  
based upon predetermined parameters defined by the user. The ASON

controller 340 provides network status information to the network management/optimization elements 380 and receives network updates from the network management/optimization elements 380 via the ASON NMI 370.

5           The OSA 610 is typically implemented in software, and can be implemented in a platform dependent or platform independent manner. In a platform dependent OSA implementation, the OSA 610 is implemented specifically for a particular platform, and is typically not portable to other platforms. In a platform independent OSA implementation, the OSA 610 is  
10 implemented so as to work with multiple platforms, for example, by separating platform-specific functions from the higher level protocols and algorithms and implementing the platform-specific functions separately from the higher level protocols and algorithms such that the higher level protocol/algorithm "layer" can be used with the platform-specific "layer" for  
15 a particular platform. The suitability of a particular programming language for implementing the OSA 610 may depend on the type of implementation (platform dependent or platform independent) as well as the execution frequency of the OSA 610. For example, a Java implementation may be suitable for a platform independent implementation in which the OSA 610 is  
20 executed infrequently, while a C++ implementation may be more suitable for platform dependent implementations as well as applications in which the OSA 610 is executed more frequently.

Whether the OSA 610 is implemented in a platform dependent or  
25 platform independent manner, the OSA 610 typically includes various user-controllable and user-customizable features. In order for a user application to access these user-controllable and user-customizable features, the OSA 610 typically includes an OSA Application Program Interface (API) that includes various primitives for accessing the user-controllable and user-customizable  
30 features of the OSA 610. The OSA API is typically simpler than the ASON UNI, particularly because the OSA API is typically an internal software



interface that does not need to implement the complexities of the ASON UNI (such as mechanisms for communicating over a UNI signaling channel).

FIG. 7 shows the relationship between a user application 710 and the OSA 610. The OSA 610 provides an OSA API through which the user application 710 can access the OSA 610. The OSA API includes various primitives for accessing the user-controllable and user-customizable features of the OSA 610.

In a typical embodiment of the OSA 610, the OSA 610 is separated into two components, namely an application component (referred to hereinafter as the OSA-A) and a network component (referred to hereinafter as the OSA-N). The OSA-A and the OSA-N communicate via a control interface, which, depending on the placement of the OSA-N (discussed below), may be the ASON UNI or another control interface.

The OSA-A is the application piece of the OSA 610. The OSA-A implements application-specific services and intelligence. Because the OSA-A is so tightly coupled to the user application, the OSA-A typically resides on the user platform in the edge system. The OSA-A manages communication services on behalf of the user, specifically by requesting communication services from the ASON 120 via the OSA-N and mapping communication services from the ASON 120 to the user network or application.

The OSA-N is the network piece of the OSA 610. Among other things, the OSA-N provides functionality for user authentication, registration, and membership. Authentication functionality enables the network to authenticate the user in order to ensure that the user is permitted to access ASON communication services. Registration functionality enables the user to register a user identifier with the network. For example, in an Internet Service Provider (ISP) application, the user identifier is typically the address of the router attachment to the ASON device (i.e., the IP address of the ASON

controller and the UNI control channel identifier), which can be viewed as the L1/L2 address of the router interface. Membership functionality enables the user to join a multicast group along with other peer users of the ASON 120. The OSA-N may reside on the user platform in the edge system or in the ASON device at the edge of the ASON 120.

FIG. 8 shows the relationship between the OSA-A 810 and the OSA-N 820. The OSA-A implements application-specific services and intelligence for managing communication services on behalf of the user. The OSA-N provides functionality for user authentication, registration, and membership. The OSA-A and the OSA-N communicate via the OSA control interface 830.

FIG. 9 shows an exemplary system in which the OSA-N 820 resides in the OSA-enabled user in the edge system. Specifically, the OSA-enabled user includes both the OSA-A 810 and the OSA-N 820. The OSA-A 810 and the OSA-N 820 communicate via the OSA control interface 830. In this case, the OSA control interface 830 is typically a software interface between the OSA-A 810 and OSA-N 820 components. The OSA-A 810 implements application-specific services and intelligence. The OSA-N 820 provides functionality for user authentication, registration, and membership, and implements the ASON UNI and other mechanisms for communicating with the ASON controller 340 in the ASON device. In this configuration, the OSA-N 820 can be implemented as a device driver for the specific UNI signaling channel interface between the OSA-enabled user and the ASON device. The OSA-A 810 can be used over different types of UNI signaling channels, for example, by installing an appropriate OSA-N device driver that supports the UNI signaling channel for a particular application.

FIG. 10 shows an exemplary system in which the OSA-N 820 resides in the ASON user at the edge of the ASON 120. Specifically, the ASON device includes the OSA-N 820 and the ASON controller 340. The OSA-A 810 in the OSA-enabled user communicates with the OSA-N 820 in the ASON device via

the ASON UNI. With the OSA-N 820 resident in the ASON device, the ASON device (i.e., the service provider) can provide advanced communication services to the OSA-enabled user that are controlled through the OSA-A 810.

5           FIG. 11 shows an exemplary system in which the OSA-N 820 resides outside of both the OSA-enabled user and the ASON device in a proxy arrangement. In this proxy arrangement, the OSA-enabled user is not required to support the ASON UNI signaling channel over the bearer channel 1110. Instead, a separate control channel 830 can be established for  
10           communication between the OSA-A 810 in the OSA-enabled user and the OSA-N 820. The OSA-A 810 sends requests to the OSA-N 820 for ASON services, and the OSA-N 820 carries out the requests using the ASON UNI. Such a proxy-based OSA/UNI can be useful in extending ASON services to legacy optical edge equipment.

15           In order to guarantee service to its users and maintain integrity of the optical core, the ASON 120 needs to provide a level of security and authentication. The fact that there is an enabled physical connection between an ASON-enabled user and the ASON device may be sufficient authentication  
20           for the ASON-enabled user to access the ASON 120. However, as ASON usage grows and OSA is deployed, the ASON 120 will need to authenticate the OSA-enabled users so that only authorized OSA-enabled users are able to access the ASON 120. Therefore, the ASON 120 will authenticate each OSA-enabled user that registers with the ASON controller.

25           In one embodiment, an authentication server is used to authenticate OSA-enabled users. The authentication server is typically a centralized database that uses an authentication protocol for authentication. The authentication protocol can use any of a variety of authentication techniques,  
30           such as challenge-based handshake authentication or simple username/password based authentication.

In another embodiment, each ASON controller 340 uses public key technology (i.e., authentication certificates) to authenticate OSA-enabled users that register with it. This requires that the OSA understand and manage authentication certificates for the OSA-enabled user.

5

Data security is a user-based feature that needs to be implemented at the edge of the enterprise network. The OSA software need not support data integrity.

10

Architecturally, the OSA-N functionality can be implemented in different ways. The suitability of a particular OSA-N architecture depends on a number of characteristics, such as such as implementation complexity, fault tolerance, UNI bandwidth utilization, UNI signaling delay, memory utilization, and computational complexity. Because OSA-N is layered on top of the ASON UNI, an important consideration is the UNI bandwidth utilization and signaling delay for OSA signaling. Latency is important because it affects the type of applications OSA can support.

15

20

A number of exemplary OSA-N architectures, namely a client-server architecture, a distributed flooding architecture, a hybrid/proxy architecture, and an ASON-coupled architecture, are discussed herein. It should be noted, however, that the present invention is in no way limited to any of the described OSA-N implementations or to any particular OSA-N implementation.

25

30

In the client-server architecture, most of the OSA-N functionality, including authentication, registration, and group membership, is handled by an optical service server (OSS), as shown in FIG. 12. The OSS maintains authentication, registration, and group membership information for multiple OSA-enabled devices. The OSA-enabled user is typically pre-configured with a group identifier. When the OSA-enabled user is attached to the ASON 120, the OSA-N sends a registration message to the OSS. The OSA-N includes its

group identifier in the registration message. The OSS stores the group identifier in its registration database. The OSA-N queries the OSS to obtain group membership information that includes the identity and location of peer users.

5

FIG. 13 is a message flow diagram depicting various exchanges between the OSA-N and the OSS in the client-server architecture. The OSA-N sends a registration message 1302 to the OSS including the group identifier (ID) for the OSA-enabled user. The OSS stores the group identifier in its registration database. The OSA-N then sends a query message 1304 to the OSS to obtain group membership information that includes the identity and location of peer users. The OSS sends peer information 1306 to the OSA-N in response to the query 1304.

10

15

The client-server architecture has a number of characteristics that are considered to be advantages. First, the client-server architecture is relatively simple to implement. Second, the amount of signaling bandwidth (both UNI and NNI) is relatively small, in part because group membership information is maintained and distributed by the OSS and therefore there is no need to advertise group membership information to the network. Signaling bandwidth can be further reduced by having the OSA-N cache the group membership information it retrieves from the OSS, with a periodic refresh technique used to keep the cache current.

20

25

The client-server architecture has a number of characteristics that are considered to be disadvantages. First, the OSS represents a single point of failure so that, if the OSS fails, the network cannot perform authentication, registration, and group membership functions. This problem can be mitigated by running a "backup" OSS and using a synchronization protocol, such as the Server Cache Synchronization Protocol (SCSP) as described in RFC 2334, to maintain synchronization between the servers, although this adds implementational complexity. Second, because the client-server

30

architecture is server-based, the network is heavily dependent upon the availability, reliability, and performance of the OSS, which could become a bottleneck for overall system performance, specifically in terms of delay. Third, like other types of centralized solutions, the client-server architecture is not scalable. Fourth, in order to register with the OSS, the OSA-enabled user must be manually configured with its group identifier.

In the distributed flooding architecture, authentication is handled by the OSS, as shown in FIG. 14. The OSS maintains authentication information for multiple OSA-enabled users, and also maintains a group identifier for each OSA-enabled user. After authenticating an OSA-enabled user, the OSS informs the OSA-enabled user of its group identifier. The OSA-enabled user then informs the other OSA-enabled users in the network of its group membership, specifically by flooding an advertisement through the network, for example, in a manner similar to OSPF-TE and PNNI Augmented Routing. This flooding occurs periodically and, since the procedure is similar to that of OSPF and PNNI, inherits the advantages of both protocols. The link state advertisement (LSA) and neighbor database is maintained in the user domain rather than in the network domain.

FIG. 15 is a message flow diagram depicting various exchanges between the OSA-N and the OSS in the distributed flooding architecture. The OSS authenticates the OSA-N through a number of exchanges 1502. After authenticating the OSA-N, the OSS determines the group identifier (ID) for the OSA-enabled user, and sends a group identifier (ID) 1504 to the OSA-N. The OSA-N sends an advertisement message 1506 into the ASON. The advertisement message 1506 is flooded throughout the network.

The distributed flooding architecture has a number of characteristics that are considered to be advantages of the distributed flooding architecture. First, it allows centralized policy as to which OSA-enabled users can join a group. Second, it does not require manual configuration of the group

identifier, and instead uses the OSS to distribute the group identifier after authentication of the OSA-enabled user. Third, it is well-suited to IP-router end systems, since OSPF is an extensible IP protocol by virtue of Opaque Link State Advertisements (LSA).

5

The distributed flooding architecture has a number of characteristics that are considered to be disadvantages of the distributed flooding architecture. First, if the OSS is out of service, then the OSA-enabled user will not be able to authenticate and determine its group identifier, and so the OSA-enabled user will not be able to determine its peers automatically. This problem can be mitigated by providing peer information to the OSA-enabled user manually, which would allow connections to be made using the ASON. In this respect, the distributed flooding architecture advantageously separates optical network operation from the availability of the OSS and allows the OSA to own most of the intelligence. Second, the flooding mechanism uses extra bandwidth in the ASON signaling network (both UNI and NNI), which can be particularly problematic if the UNI has limited bandwidth. Third, flooding does not occur until authentication is complete. Fourth, an OSPF-based flooding mechanism requires IP support, and is therefore not suitable for non-IP routers. Fifth, because the LSA and neighbor database is maintained in the user domain rather than the network domain, topological information must be "leaked" to the OSA-enabled user, which can be done by implementing some form of NNI in the UNI and thereby confusing the separation between the NNI and the UNI.

25

In the hybrid/proxy architecture, authentication and flooding are handled by the OSS, as shown in FIG. 16. The OSS maintains authentication information for multiple OSA-enabled users, and also maintains a group identifier for each OSA-enabled user. After authenticating an OSA-enabled user, the OSS floods the advertisement on behalf of the OSA-enabled user, for example, using a mechanism similar to Proxy-PAR.

30

FIG. 17 is a message flow diagram depicting various exchanges between the OSA-N and the OSS in the hybrid/proxy architecture. The OSS authenticates the OSA-N through a number of exchanges 1702. After authenticating the OSA-N, the OSS determines the group identifier (ID) for the OSA-enabled user, and sends an advertisement message 1706 into the ASON. The advertisement message 1706 is flooded throughout the network.

The hybrid/proxy architecture is similar to the distributed flooding architecture, and therefore has many of the same advantageous and disadvantageous characteristics as the distributed flooding architecture. However, because the OSS rather than the OSA-N floods the advertisement, the OSS does not need to "leak" topological information to the OSA-enabled user. Therefore, there is no confusion of the separation between the NNI and the UNI.

In the ASON-coupled architecture, the various OSS services described with reference to the client-server architecture are handled by the ASON devices at the periphery of the ASON 120, with each ASON device acting as the OSS for its directly attached OSA-enabled device. The ASON-coupled architecture has many characteristics that are considered to be advantages of the ASON-coupled architecture. First, because there is no central server, network performance does not depend upon the availability and reliability of a single server. Second, because an ASON device services only its directly attached OSA-enabled user, a failure of the ASON device affects only the directly attached OSA-enabled device. Third, the ASON-coupled architecture is scalable, particularly because of the one-to-one mapping between clients and servers, and also because the ASON device only needs to maintain advertisements for the group associated with its directly attached OSA-enabled user. Fourth, the amount of UNI signaling bandwidth is relatively small. Fifth, because link state advertisements are stored by the ASON device rather than in the user domain, the ASON device can detect a failure of its directly attached OSA-enabled user and invalidate the link state



advertisement for the OSA-enabled user, thus allowing peer users to more quickly detect the failure of the OSA-enabled user (otherwise, the peer users would need to wait for a link state advertisement timeout to detect the failure, which, in OSPF, is typically a MaxAge of 1 hour). Sixth, because the ASON device is already running ASON-specific OSPF, adding extra link state advertisements into the link state database is relatively easy. Seventh, because no topological information needs to be "leaked" to the OSA-enabled user, the separation of UNI and NNI is clear. Eighth, because the link state advertisements are handled by the ASON device, the OSA-enabled user does not necessarily need to implement OSPF, which can be a big advantage in certain applications.

Of the four OSA-N implementation architectures described above, the ASON-coupled architecture seems most suitable for typical applications, particularly because of its clearly defined separation of UNI and NNI, flexible implementation platforms, and simplicity for client systems. OSA-A must be customized on a case-by-case basis and requires a lot of close work with customer.

As discussed above, both the ASON-enabled user and the OSA-enabled user implement at least some of the ASON UNI functionality for obtaining communication services from the ASON 120. The OSA-enabled user additionally includes the OSA 610 for performing a variety of advanced communication services using the ASON UNI.

As shown and discussed with reference to FIG. 5 above, two ASON-enabled users can communicate over the ASON 120 using the ASON UNI. However, the types of communication services available to the ASON-enabled users is essentially limited to those that are provided directly by the ASON 120.

On the other hand, an OSA-enabled user can interoperate with ASON-enabled users and/or with other OSA-enabled users over the ASON 120.

When interoperating with other OSA-enabled users, a full set of OSA-enabled features can be supported end-to-end across the ASON 120. When

5 interoperating with an ASON-enabled user that does not support OSA, only a limited set of features can be supported end-to-end across the ASON 120 (such as setting up an optical communication path through the ASON 120).

FIG. 18 shows an exemplary communication system 1800 in which an  
10 OSA-enabled user 1810 communicates with an ASON-enabled user 1820 over the ASON 120. The ASON-enabled user 1820 implements some or all of the ASON UNI. The OSA-enabled user 1810 implements advanced functions using the ASON UNI. Because the ASON-enabled user 1820 does not  
15 implement OSA functionality, the OSA-enabled user 1810 and the ASON-enabled user 1820 can support only a limited set of features end-to-end across the ASON 120. For example, using the ASON UNI, the OSA-enabled user 1810 and the ASON-enabled user 1820 can establish an end-to-end optical communication path for communicating over the ASON 120.

FIG. 19 shows an exemplary communication system 1900 in which two  
20 OSA-enabled users 1910, 1920 communicate over the ASON 120. The OSA-enabled users 1910, 1920 implement advanced functions using the ASON UNI. Because both OSA-enabled users 1910, 1920 implement OSA functionality, the OSA-enabled users 1910, 1920 can support a full set of  
25 features end-to-end across the ASON 120.

In order to manage various communication services, the OSA 610  
includes optical service logic that implements application-specific services and intelligence. The optical service logic interacts with the ASON 120 via the  
30 ASON UNI. The optical service logic also interacts with other OSA-enabled users via a peer-to-peer signaling mechanism. The peer-to-peer signaling mechanism enables communication between OSA-enabled users within a user

network and/or across the ASON 120. Thus, using the peer-to-peer signaling mechanism, OSA-enabled users within the user network can interoperate over intra-domain signaling channels, and OSA-enabled users at the edge of the ASON 120 can interoperate across the ASON 120 over ASON signaling channels. Among other things, the peer-to-peer signaling mechanism effectively extends OSA functionality to OSA-enabled users that are not at the edge of the ASON 120.

It should be noted that, while the OSA-enabled users at the edge of the ASON 120 implement the ASON UNI and interact with the ASON 120 via the ASON UNI, OSA-enabled users that are not adjacent to the ASON 120 are unable to interact directly with the ASON 120 via the ASON UNI. Therefore, such OSA-enabled users cannot directly access ASON services. Such OSA-enabled users can, however, channel ASON service requests through the OSA-enabled user at the edge of the ASON 120 using the peer-to-peer signaling mechanism. Specifically, an OSA-enabled user forwards an ASON service request to an OSA-enabled user at the edge of the ASON 120 using the peer-to-peer signaling mechanism. In turn, the OSA-enabled user at the edge of the ASON 120 interacts with the ASON 120 via the ASON UNI to execute the ASON service request, and responds if necessary using the peer-to-peer signaling mechanism.

Certain communication services can be managed by a single OSA-enabled user. However, other communication services require that a number of OSA-enabled users interoperate to coordinate communication services. This is particularly true when communication services are coordinated end-to-end between peer OSA-enabled users across the ASON 120. In order for OSA-enabled users to interoperate, each OSA-enabled user needs to identify its peer OSA-enabled users and obtain various types of peer information for each peer OSA-enabled user, such as an attachment address that is used for establishing an optical communication path to the peer OSA-enabled user.

Therefore, the OSA typically includes mechanisms for identifying peer OSA-enabled users and for obtaining the peer information.

In a typical embodiment of the present invention, the OSA includes  
5 auto-discovery logic by which an OSA-enabled user automatically discovers  
its peer OSA-enabled users and obtains the various types of peer information  
for each of its peer OSA-enabled users. The auto-discovery logic typically  
uses an advertisement mechanism for exchanging peer information between  
OSA-enabled users, similar to the exchanging of link state advertisements by  
10 OSPF, although the auto-discovery logic is not limited to any particular  
advertisement or discovery mechanism. Each OSA-enabled user typically  
includes a peer database in which the peer information is stored. It should be  
noted that peer information can also be configured manually, for example, by  
a network administrator.

In addition to identifying peer OSA-enabled users, it is typically  
15 necessary or desirable for each OSA-enabled user to authenticate its peers.  
Peer authentication is important because OSA operations can affect such  
things as the integrity of the ASON 120 and of the network as a whole.  
20 Therefore, the OSA typically includes peer authentication logic for  
authenticating peer OSA-enabled users. The peer authentication logic  
typically uses public or private key technologies for authentication, although  
the peer authentication logic is not limited to any particular peer  
authentication mechanism.

25 FIG. 20 shows the relevant components of the OSA 610. Among other  
things, the OSA 610 includes network management logic 2010, optical service  
logic 2020, peer database 2030, auto-discovery logic 2040, ASON UNI 2050,  
peer-to-peer signaling logic 2060, and peer authentication logic 2070.

30 The network management logic 2010 provides for configuration and  
control of the OSA 610. Among other things, the network management logic

2010 interfaces with the network management/optimization elements 380 in the ASON device via the network management interface 620, and also enables remote control of the OSA 610 by a network administrator. For example, the network administrator can manually configure peer information in the peer database 2030 via the network management logic 2010.

The optical service logic 2020 implements application-specific services and intelligence. The optical service logic 202 interacts with the ASON 120 via the ASON UNI 2050. The optical service logic 2020 also interacts with other OSA-enabled users via the peer-to-peer signaling logic 2060. The optical service logic 2020 may utilize peer information stored in the peer database 2030.

The peer-to-peer signaling logic 2060 enables the OSA-enabled user to communicate with other OSA-enabled users within a user network and/or across the ASON 120. Among other things, the peer-to-peer signaling mechanism effectively extends OSA functionality to OSA-enabled users that are not at the edge of the ASON 120. The peer-to-peer signaling logic 2060 may utilize peer information stored in the peer database 2030.

The auto-discovery logic 2040 enables the OSA-enabled user to automatically discover peer OSA-enabled users within a user network and/or across the ASON 120. The auto-discovery logic 2040 typically uses an advertisement mechanism for exchanging peer information between OSA-enabled users, similar to the exchanging of link state advertisements by OSPF, although the auto-discovery logic 2040 is not limited to any particular advertisement or discovery mechanism. The auto-discover logic 2040 stores peer information in the peer database 2030.

The peer authentication logic 2070 enables the OSA-enabled user to authenticate peer OSA-enabled users. Peer authentication is important because OSA operations can affect such things as the integrity of the ASON

120 and of the network as a whole. The peer authentication logic 2070 typically uses public or private key technologies for authentication, although the peer authentication logic 2070 is not limited to any particular peer authentication mechanism. The peer authentication logic 2070 may utilize  
5 and store peer information in the peer database 2030.

FIG. 21 shows exemplary OSA logic 2100 for managing communication services. Beginning at block 2102, the logic discovers peer OSA-enabled users using a predetermined auto-discovery mechanism, in block 2104. The logic  
10 authenticates the peer OSA-enabled users using a predetermined peer authentication mechanism, in block 2106. The logic then interoperates with the peer OSA-enabled users to manage and coordinate communication services, in block 2108. The logic 2100 terminates in block 2199.

15 In an exemplary embodiment of the present invention, an authenticated auto-discovery mechanism combining both auto-discovery and peer authentication is used to automatically identify and authenticate peer OSA-enabled users. The authenticated auto-discovery mechanism requires each OSA-enabled user to register with the ASON using an authenticated  
20 registration mechanism. A centralized advertisement scheme managed by the OSS is used to collect and distribute peer information to the peer OSA-enabled users associated with a particular peer group. Each OSA-enabled user maintains the peer information received from the OSS in its peer database.

25 More specifically, when an OSA-enabled user needs to access the ASON, it first establishes and activates the ASON UNI with a corresponding ASON device at the edge of the ASON. The OSA-enabled user then registers with the ASON by sending a registration message to the ASON edge device.  
30 The registration message includes, among other things, a group identifier identifying the peer group for the OSA-enabled user.

Upon receiving the registration message from the OSA-enabled user, the ASON edge device sends a challenge message to the OSA-enabled user. The challenge message provides an opportunity for the OSA-enabled user to positively identify itself to the ASON through a cryptographic authentication mechanism, for example, using predetermined public and/or private key technologies.

Upon receiving the challenge message from the ASON edge device, the OSA-enabled user formats a challenge response message. The challenge response message positively identifies itself to the ASON device using the cryptographic authentication message. The OSA-enabled user sends the challenge response message to the ASON edge device.

Upon receiving the challenge response message from the OSA-enabled user, the ASON edge device authenticates the information in the challenge response message to verify and positively identify the OSA-enabled user. This authentication may require interaction with other network elements, such as a certifying authority for public key authentication or retrieval of an encryption key from a secure server (possibly the OSS) for private key authentication. If the ASON edge device is able to verify and positively identify the OSA-enabled user through the information provided in the challenge response message, then the ASON edge device sends a success message to the OSA-enabled user indicating that the registration process is complete. On the other hand, if the ASON edge device is unable to verify and positively identify the OSA-enabled user through the information provided in the challenge response message, then the ASON edge device rejects the registration, for example, by sending a rejection message to the OSA-enabled user.

Upon successfully registering the OSA-enabled user, the ASON edge device also sends a join message to the OSS in order to add the OSA-enabled user to its peer group. The join message includes, among other things, a

group identifier identifying the peer group, a user identifier identifying the OSA-enabled user, and a bearer identifier identifying the bearer channel associated with the OSA-enabled user.

5           The OSS maintains group membership information for the various OSA-enabled users that have registered with the ASON. Upon receiving the join message from the ASON edge device, the OSS stores the group membership information for the new OSA-enabled user identified in the join message. If the new OSA-enabled user is the first to register for the particular  
10 peer group, then the OSS sends a database synchronization message to the ASON edge device listing no peer OSA-enabled users (i.e., a NULL list). However, if the new OSA-enabled user is not the first to register for the particular peer group, then the OSS sends a database synchronization message to the ASON edge device listing the other peer OSA-enabled users,  
15 and also sends an advertisement message to the various ASON devices supporting registered OSA-enabled users listing at least the new OSA-enabled user.

          Upon receiving the database synchronization message from the OSS,  
20 the ASON edge device determines whether any peer OSA-enabled users are listed in the database synchronization message. If there is at least one peer OSA-enabled user listed in the database synchronization message, then the ASON edge device sends a new neighbor message to the OSA-enabled user listing the peer OSA-enabled users and their respective bearer identifiers.  
25 However, if there are no peer OSA-enabled users listed in the database synchronization message (i.e., the list is NULL), then the ASON edge device typically does not send the new neighbor message to the OSA-enabled user, since the peer database maintained by the OSA-enabled user is NULL by default.

30           Each ASON device that receives an advertisement message from the OSS sends a new neighbor message to its respective OSA-enabled user



including the list of peer OSA-enabled users from the advertisement message. The new neighbor message identifies the new OSA-enabled user to all existing OSA-enabled users in the peer group.

5           Upon receiving a new neighbor message from its corresponding ASON device, an OSA-enabled user stores the peer information from the new neighbor message in its peer database.

10           Thereafter, whenever a new OSA-enabled user registers with the peer group, the OSS sends an advertisement message to those ASON devices that support registered OSA-enabled devices in the peer group. The advertisement message identifies at least the new OSA-enabled user and its bearer identifier, and may also list some or all of the other peer OSA-enabled users and their respective bearer identifiers. Each ASON device that receives  
15           an advertisement message from the OSS sends a new neighbor message to its respective OSA-enabled user listing the peer OSA-enabled users and their respective bearer identifiers. Each OSA-enabled user that receives a new neighbor message from its corresponding ASON device stores the peer information in its peer database.

20

FIG. 22 is a message flow diagram demonstrating the authenticated auto-discovery process between an OSA-enabled user A and an OSA-enabled user B. The OSA-enabled user A accesses the ASON through ASON device O1. The OSA-enabled user B accesses the ASON through ASON device O2.

25           In this example, it is assumed that the OSA-enabled user A is the first to register for the peer group G, and the OSA-enabled user B is the second to register for the peer group G.

30           In order to register with the ASON, the OSA-enabled user A sends registration message 2202 to the ASON device O1 indicating peer group G. The ASON device O1 sends challenge message 2204 to the OSA-enabled user A. The OSA-enabled user A sends challenge response message 2206 to the

ASON device O1. The ASON device O1 sends success message 2208 to the OSA-enabled user A, and also sends join message 2210 to the OSS including group identifier for peer group G, user identifier for OSA-enabled user A, and bearer identifier for the bearer channel to OSA-enabled user A. The OSS  
5 sends database synchronization (DBsync) message 2212 to the ASON device O1 with a NULL list of peer OSA-enabled users. The ASON device O1 does not send a new neighbor message to the OSA-enabled user A.

In order to register with the ASON, the OSA-enabled user B sends  
10 registration message 2214 to the ASON device O2 indicating peer group G. The ASON device O2 sends challenge message 2216 to the OSA-enabled user B. The OSA-enabled user B sends challenge response message 2218 to the ASON device O2. The ASON device O2 sends success message 2220 to the OSA-enabled user B, and also sends join message 2222 to the OSS including  
15 group identifier for peer group G, user identifier for OSA-enabled user B, and bearer identifier for the bearer channel to OSA-enabled user B. The OSS sends database synchronization (DBsync) message 2228 to the ASON device O2 with listing OSA-enabled user A as a peer OSA-enabled user, and also sends advertisement message 2226 to the ASON device O1 indicating OSA-  
20 enabled user B as a new peer OSA-enabled user. The ASON device O1 sends new neighbor message 2224 to the OSA-enabled user A including peer information for OSA-enabled user B, and the OSA-enabled user A adds OSA-enabled user B to its peer database. The ASON device O2 sends new  
25 neighbor message 2230 to the OSA-enabled user B including peer information for OSA-enabled user A, and the OSA-enabled user B adds OSA-enabled user A to its peer database. At this point, the OSA-enabled user A has successfully identified and authenticated the OSA-enabled user B, and the OSA-enabled user B has successfully identified and authenticated the OSA-enabled user A.

30 Each ASON device monitors the connection to its corresponding OSA-enabled user. If the ASON device detects loss or degradation of the connection to the OSA-enabled user (e.g., due to a failure of the ASON UNI,

the bearer channel, or the OSA-enabled device itself), then the ASON device sends a leave message to the OSS to remove the OSA-enabled user from the peer group. The OSS removes the OSA-enabled user from the peer group, and sends an advertisement message to the various ASON devices that support registered OSA-enabled users indicating the removed OSA-enabled users. Each ASON device in turn sends an update message to its corresponding OSA-enabled user indicating the removed OSA-enabled user. Each OSA-enabled user deletes the removed OSA-enabled user from its respective peer database.

FIG. 23 is a message flow diagram demonstrating the process of removing an OSA-enabled user from a peer group. Upon detecting loss or degradation of the connection to the OSA-enabled user, as represented by 2302, the ASON device O1 sends leave message 2304 to the OSS. The OSS removes the OSA-enabled device A from the peer group, and sends advertisement message 2306 to the ASON device O2. The ASON device O2 sends update message 2308 to the OSA-enabled user B indicating that the OSA-enabled user A is no longer a member of the peer group, and the OSA-enabled user B removes the OSA-enabled user A from its peer database.

It should be noted that, when an OSA-enabled user interoperates with an ASON-enabled user, as shown and described with reference to FIG. 18 above, there is essentially no peer-to-peer relationship between the OSA-enabled user and the ASON-enabled user. Consequently, many of the OSA's peer-to-peer mechanisms are not available for interoperation between the OSA-enabled user and the ASON-enabled user. Specifically, the ASON-enabled user does not support OSA auto-discovery, peer authentication, and peer-to-peer signaling mechanisms. Therefore, the OSA-enabled user typically cannot auto-discover, authenticate, and perform peer-to-peer signaling with the ASON-enabled user. This has certain practical implications. For example, an ASON-enabled user receiving a request from an OSA-enabled user would be unable to authenticate the OSA-enabled user,

and therefore would generally have to accept the request unconditionally. This poses a security/reliability risk that may be unacceptable in some situations.

5           It should also be noted that optical communication services may be provided by multiple service and infrastructure providers. The OSA can manage communication services provided by these multiple service and infrastructure providers.

10           As discussed above, the OSA is an intelligent agent that manages various communication services on behalf of the network user. The OSA interacts with the ASON to obtain various communication services, and manages those services for the network user based upon predetermined parameters defined by the network user. The OSA can manage practically  
15 any communication services that have heretofore been managed manually.

The related application entitled **SYSTEM, DEVICE, AND METHOD FOR MANAGING CONNECTION ESTABLISHMENT AND RELATED SERVICES IN AN OPTICAL COMMUNICATION SYSTEM**, which was  
20 incorporated by reference above, discusses various connection establishment services that can be managed by the OSA. For example, the OSA can negotiate various connection and connection-related services on behalf of the user, model one or more connections for the user, reserve connection and connection-related services for the user, establish a connection for the user,  
25 and aggregate multiple optical communication paths over a connection, to name but a few.

When a connection is established, a service level agreement (SLA) is often established between the network user and a service provider, which can  
30 be the ASON and/or a service provider associated with the ASON. The SLA is essentially an agreement by the service provider to provide a specific level of service to the network user. The SLA may specify such things as the

bandwidth, quality (e.g., error rate, dropped packet rate, latency), duration, protection class, and protection/failover mechanism for the connection, to name but a few. The SLA typically provides for various penalties in case the service provider is unable to provide the level of service agreed to in the SLA.

5

In a perfect world, the service provider would always provide a level of service specified by the SLA. Unfortunately, the service provider is not always able to provide the specified level of service, for example, due to network failures, congestion, overbooking of network resources, and other  
10 circumstances. Therefore, it is common for the user to monitor the connection to identify any breaches of the SLA and invoke the penalty provisions of the SLA. Some breaches of the SLA are obvious, such as a complete failure of a connection. However, other breaches of the SLA are less obvious. Thus, the network user must typically police the SLA in order to identify breaches of  
15 the SLA. This is typically done manually, for example, by monitoring the connection or through off-line analysis of statistical information maintained by the network user. This policing of the SLA by the network user is difficult, time-consuming, error-prone, and, in most cases, untimely.

20

Therefore, in an embodiment of the present invention, the OSA manages SLAs on behalf of the OSA-enabled user. The OSA can perform both real-time and off-line analysis, and can interact with various network elements (including the ASON) to handle billing, penalty, and other issues associated with a SLA breach.

25

For example, the OSA may monitor the connection for real-time analysis of SLA compliance on behalf of the OSA-enabled user. Specifically, the OSA may monitor the integrity of the connection as well as the traffic on the connection to verify that the connection meets certain SLA criteria. The  
30 OSA may also query the ASON via the ASON UNI in order to obtain information compiled by the ASON that the OSA can use to verify the SLA. The OSA may also query peer OSA-enabled users via the peer-to-peer

interface in order to obtain information compiled by the peer OSA-enabled users (which may also monitor, query, and verify SLA) that the OSA can use to verify the SLA.

5           The OSA may also gather and maintain statistical information for off-line analysis of SLA compliance on behalf of the OSA-enabled user. Specifically, the OSA may store statistical information it obtains from monitoring the connection, querying the ASON, and querying the peer OSA-enabled users, such as the number of rejected calls and the number of  
10       dropped packets. The OSA can then use the stored statistical information for off-line analysis. Such off-line analysis can be used not only to detect breaches of SLA, but also to identify patterns or trends that relate to SLA. For example, the off-line analysis might identify a time of day at which the SLA tends to breach the SLA on a regular basis.

15           The OSA may also interact with the service provider to enforce penalty provisions in the SLA on behalf of the OSA-enabled user. Specifically, the OSA may notify the service provider of SLA breaches and invoke appropriate penalty provisions in the SLA. This can be done in real-time or at any time  
20       after a breach is detected.

          The OSA may also interact with the service provider to negotiate a credit for services not provided by the service provider. Specifically, the user may be entitled to a credit when the service provider fails to provide the level  
25       of service specified in the SLA. For example, if the actual duration of the connection was half of the agreed upon duration, then the user may be entitled to a credit of half the cost of the connection. The OSA may negotiate a credit in real-time or at any time after a breach is detected.

30           The OSA may also interact with the service provider to negotiate "replacement" services on behalf of the OSA-enabled user. Specifically, the user may accept additional services from the service provide in lieu of, or in

addition to, a penalty or credit. For example, if the service provider did not provide sufficient bandwidth during a particular period, then the OSA may accept additional bandwidth later in lieu of a credit.

5           The OSA may also interact with various network elements to rectify a breach of the SLA on behalf of the OSA-enabled user. Specifically, the OSA may work unilaterally or in conjunction with service provider and/or the peer OSA-enabled users to resolve or work around certain network problems. For example, when a connection fails to meet the SLA, the OSA may re-  
10   request the connection in order to force the service provider to provision a new connection that meets the SLA. Alternative, the OSA may notify the service provider of the breach and orchestrate various network changes to resolve or work around the breach.

15           The OSA may also interact with the service provider to dynamically modify the SLA on behalf of the OSA-enabled user. Specifically, the OSA may dynamically re-negotiate the SLA based upon changing requirements of the OSA-enabled user. For example, the OSA may dynamically modify the SLA for such things as changing the duration of the connection, changing the  
20   bandwidth of the connection, changing the quality of the connection, establishing/ changing protection/ failover mechanism, to name but a few. There may be additional charges for dynamically modifying the SLA and/or for various changes associated with the modified SLA (e.g., additional connection duration, additional connection bandwidth, improved connection  
25   quality, protection/ failover).

          The OSA may also interface with a billing/ accounting system to provide SLA-related information on behalf of the OSA-enabled user. Specifically, the OSA may provide various types of information to the  
30   billing/ accounting system, such as usage information, penalty information, and information regarding credits or additional charges. The billing/ accounting system can use this information, for example, to check for

billing errors, verify accounts payable information, or compute accounts payable information, to name but a few. Such a value-added service is particular useful, since otherwise the OSA-enabled user has virtually no way to verify and track usage charges, penalties, credits, and additional charges based upon SLA.

FIG. 24 is a logic flow diagram showing exemplary OSA logic 2400 for managing SLA on behalf of the network user. Beginning at block 2402, the OSA may monitor the connection for real-time analysis of SLA compliance on behalf of the OSA-enabled user, in block 2404. The OSA may gather and maintain statistical information for off-line analysis of SLA compliance on behalf of the OSA-enabled user, in block 2406. The OSA may interact with the service provider to enforce penalty provisions in the SLA on behalf of the OSA-enabled user, in block 2408. The OSA may interact with the service provider to negotiate a credit for services not provided by the service provider, in block 2410. The OSA may interact with the service provider to negotiate "replacement" services on behalf of the OSA-enabled user, in block 2412. The OSA may interact with various network elements to rectify a breach of the SLA on behalf of the OSA-enabled user, in block 2414. The OSA may interact with the service provider to dynamically modify the SLA on behalf of the OSA-enabled user, in block 2416. The OSA may interface with a billing/accounting system to provide SLA-related information on behalf of the OSA-enabled user, in block 2418. The logic 2400 terminates in block 2499.

FIG. 25 is a logic flow diagram showing exemplary OSA logic 2500 for verifying the SLA and identifying breaches of the SLA. Beginning at block 2502, the OSA may monitor the integrity of the connection to verify that the connection meets certain SLA criteria, in block 2504. The OSA may monitor traffic on the connection to verify that the connection meets certain SLA criteria, in block 2506. The OSA may query the ASON and/or peer OSA-enabled users to obtain connection-related information, in block 2508. The OSA may verify the SLA in real-time using the connection-related



information obtained from the ASON and/or the peer OSA-enabled users, in block 2510. The OSA may store the connection-related information obtained from the ASON and/or the peer OSA-enabled users, in block 2512, and analyze the stored connection-related information off-line to detect breaches, patterns, and trends that relate to SLA, in block 2514. The logic 2500 terminates in block 2599.

FIG. 26 is a logic flow diagram showing exemplary OSA logic 2600 for managing the SLA upon detecting a breach of the SLA. Beginning at block 2602, and upon detecting a breach of the SLA, in block 2604, the OSA may notify the service provider of the SLA breach, in block 2606. The OSA may interact with the service provider to invoke appropriate penalty provisions in the SLA for the SLA breach, in block 2608. The OSA may negotiate with the service provider for a credit, in block 2610. The OSA may negotiate with the service provider for additional services in lieu of, or in addition to, a penalty or credit, in block 2612. The OSA may work unilaterally or in conjunction with service provider and/or the peer OSA-enabled users to resolve or work around certain network problems, in block 2614. The OSA may dynamically re-negotiate the SLA based upon changing requirements of the OSA-enabled user, in block 2616. The logic 2600 terminates in block 2699.

It should be noted that the term "router" is used herein to describe a communication device that may be used in a communication system, and should not be construed to limit the present invention to any particular communication device type. Thus, a communication device may include, without limitation, a bridge, router, bridge-router (brouter), switch, node, or other communication device.

It should also be noted that the term "packet" is used herein to describe a communication message that may be used by a communication device (*e.g.*, created, transmitted, received, stored, or processed by the communication device) or conveyed by a communication medium, and should not be

construed to limit the present invention to any particular communication message type, communication message format, or communication protocol. Thus, a communication message may include, without limitation, a frame, packet, datagram, user datagram, cell, or other type of communication message.

It should also be noted that the logic flow diagrams are used herein to demonstrate various aspects of the invention, and should not be construed to limit the present invention to any particular logic flow or logic implementation. The described logic may be partitioned into different logic blocks (e.g., programs, modules, functions, or subroutines) without changing the overall results or otherwise departing from the true scope of the invention. Often times, logic elements may be added, modified, omitted, performed in a different order, or implemented using different logic constructs (e.g., logic gates, looping primitives, conditional logic, and other logic constructs) without changing the overall results or otherwise departing from the true scope of the invention.

The present invention may be embodied in many different forms, including, but in no way limited to, computer program logic for use with a processor (e.g., a microprocessor, microcontroller, digital signal processor, or general purpose computer), programmable logic for use with a programmable logic device (e.g., a Field Programmable Gate Array (FPGA) or other PLD), discrete components, integrated circuitry (e.g., an Application Specific Integrated Circuit (ASIC)), or any other means including any combination thereof. In a typical embodiment of the present invention, predominantly all of the OSA logic is implemented as a set of computer program instructions that is converted into a computer executable form, stored as such in a computer readable medium, and executed by a microprocessor within the OSA-enabled user under the control of an operating system.

Computer program logic implementing all or part of the functionality previously described herein may be embodied in various forms, including, but in no way limited to, a source code form, a computer executable form, and various intermediate forms (*e.g.*, forms generated by an assembler, compiler, linker, or locator). Source code may include a series of computer program instructions implemented in any of various programming languages (*e.g.*, an object code, an assembly language, or a high-level language such as Fortran, C, C++, JAVA, or HTML) for use with various operating systems or operating environments. The source code may define and use various data structures and communication messages. The source code may be in a computer executable form (*e.g.*, via an interpreter), or the source code may be converted (*e.g.*, via a translator, assembler, or compiler) into a computer executable form.

The computer program may be fixed in any form (*e.g.*, source code form, computer executable form, or an intermediate form) either permanently or transitorily in a tangible storage medium, such as a semiconductor memory device (*e.g.*, a RAM, ROM, PROM, EEPROM, or Flash-Programmable RAM), a magnetic memory device (*e.g.*, a diskette or fixed disk), an optical memory device (*e.g.*, a CD-ROM), a PC card (*e.g.*, PCMCIA card), or other memory device. The computer program may be fixed in any form in a signal that is transmittable to a computer using any of various communication technologies, including, but in no way limited to, analog technologies, digital technologies, optical technologies, wireless technologies (*e.g.*, Bluetooth), networking technologies, and internetworking technologies. The computer program may be distributed in any form as a removable storage medium with accompanying printed or electronic documentation (*e.g.*, shrink wrapped software), preloaded with a computer system (*e.g.*, on system ROM or fixed disk), or distributed from a server or electronic bulletin board over the communication system (*e.g.*, the Internet or World Wide Web).

Hardware logic (including programmable logic for use with a programmable logic device) implementing all or part of the functionality

previously described herein may be designed using traditional manual methods, or may be designed, captured, simulated, or documented electronically using various tools, such as Computer Aided Design (CAD), a hardware description language (*e.g.*, VHDL or AHDL), or a PLD programming language (*e.g.*, PALASM, ABEL, or CUPL).

Programmable logic may be fixed either permanently or transitorily in a tangible storage medium, such as a semiconductor memory device (*e.g.*, a RAM, ROM, PROM, EEPROM, or Flash-Programmable RAM), a magnetic memory device (*e.g.*, a diskette or fixed disk), an optical memory device (*e.g.*, a CD-ROM), or other memory device. The programmable logic may be fixed in a signal that is transmittable to a computer using any of various communication technologies, including, but in no way limited to, analog technologies, digital technologies, optical technologies, wireless technologies (*e.g.*, Bluetooth), networking technologies, and internetworking technologies. The programmable logic may be distributed as a removable storage medium with accompanying printed or electronic documentation (*e.g.*, shrink wrapped software), preloaded with a computer system (*e.g.*, on system ROM or fixed disk), or distributed from a server or electronic bulletin board over the communication system (*e.g.*, the Internet or World Wide Web).

The present invention may be embodied in other specific forms without departing from the true scope of the invention. The described embodiments are to be considered in all respects only as illustrative and not restrictive.